

مخطط الكائنات العلائقية :-

Entity Relationship Diagram (ERD) :- هو مخطط يستخدم في تصميم قواعد البيانات لتمثيل العلاقات بين الكائنات المختلفة في النظام.

تفيد مخططات الكائنات العلائقية في تصميم قواعد البيانات لأنها تساعد على فهم العلاقات بين الكائنات والخصائص التي تميز كل منها، وتمثل بشكل بسيط وسهل الفهم للمستخدمين والمطورين. كما يمكن استخدامها لتوثيق النظام وتحديد متطلباته .

يتألف مخطط الكائنات العلائقية من ثلاثة عناصر رئيسية:

1. الكائنات (Entities) :-

هي المكونات الرئيسية في النظام، وتمثل الأشياء التي يجب تتبعها في النظام. وعادة ما يتم تمثيل الكائنات في المخطط بصورة مستطيلات مع اسم الكائن داخلها.

في مخطط الكائنات العلائقية، يتم تمثيل البيانات الموجودة في النظام عن طريق الكائنات. تتميز الكائنات بأنها تمثل الأشياء التي يجب تتبعها في النظام، وهي تشمل الأشخاص والأماكن والمنتجات والعمليات والأحداث وغيرها.

يمكن تقسيم الكائنات في مخطط الكائنات العلائقية إلى نوعين رئيسيين:-

1. الكائنات القوية : - وهي الكائنات التي يتم تصميم النظام حولها، وتشكل العناصر الرئيسية للنظام. ويمثل كل منها جزءًا مهمًا من النظام، ويمكن أن يكون لكل كائن رئيسي خصائص (Attributes) مختلفة. وتشمل الكائنات الرئيسية في مخطط الكائنات العلائقية .

2. الكائنات الضعيفة (Sub-Entities): وهي الكائنات التي يمكن تصميمها على أساس الكائنات الرئيسية، ويتم استخدامها لتوضيح وتفصيل العناصر الرئيسية. وغالبًا ما تحتوي الكائنات الثانوية على خصائص إضافية تتعلق بالعناصر الرئيسية، ويمكن استخدامها لتوضيح العلاقات بين الكائنات. ويمكن أن تشمل الكائنات الضعيفة في مخطط الكائنات العلائقية .

يمكن التفريق بين الكائنات الرئيسية والكائنات الثانوية في مخطط الكائنات العلائقية بأن الكائنات الرئيسية هي العناصر الرئيسية التي يتم تصميم النظام حولها، في حين أن الكائنات الثانوية هي العناصر التي يتم استخدامها لتوضيح وتفصيل العناصر الرئيسية. كما أن الكائنات الرئيسية عادة ما تحتوي على سمات مختلفة تميز كل كائن، بينما تحتوي الكائنات الثانوية على سمات إضافية تتعلق بالعناصر الرئيسية.

2. العلاقات (Relationships) :-

هي الروابط التي تربط الكائنات مع بعضها البعض. وتمثل العلاقات في المخطط عادة بصورة معين يربط بين اثنين من الكائنات .

العلاقات في مخطط الكائنات العلاقية تمثل الروابط بين الكائنات أو الأشياء في النظام. وتعد العلاقات أساسية في تصميم قواعد البيانات لأنها تساعد في تحديد كيفية ارتباط الكائنات ببعضها البعض .

ويمكن تقسيم العلاقات في مخطط الكائنات العلاقية إلى ثلاث أنواع رئيسية :-

1. العلاقة الواحد-إلى-واحد (1:1): وتحدث هذه العلاقة عندما يكون هناك ارتباط واحد فقط بين كائنين. على سبيل المثال، يمكن أن يكون لدى شخص واحد فقط جواز سفر واحد.
2. العلاقة الواحد-إلى-كثير (1:Many): وتحدث هذه العلاقة عندما يكون هناك ارتباط واحد بين كائن وعدة كائنات أخرى. على سبيل المثال، يمكن أن يكون لدى شخص عدة حسابات مصرفية.
3. العلاقة كثير-إلى-كثير (Many-to-Many): وتحدث هذه العلاقة عندما يكون هناك ارتباطات متعددة بين كائن وعدة كائنات أخرى. على سبيل المثال، يمكن أن يكون لدى عدة أشخاص مشتركون في عدة مجموعات.

3. السمات (Attributes) :-

هي الخصائص التي تميز كل كائن في النظام. وتمثل السمات في المخطط عادة بشكل بيضاوي مع أسم الخاصية .

تتميز الخصائص بأنها تمثل البيانات التي تحتويها الكائنات، وتشمل الخصائص في مخطط العلاقات :-

1. الخصائص البسيطة (Simple Properties): هي الخصائص التي تحتوي على قيمة واحدة فقط، مثل النص والتاريخ والرقم .
2. الخصائص المركبة (Compound Properties): هي الخصائص التي تحتوي على أكثر من قيمة، وتتألف من خصائص بسيطة أخرى، مثل العنوان والاسم الكامل.
3. الخصائص المشتقة (Derived Properties): هي الخصائص التي يتم استنتاجها من خصائص أخرى في الجدول، مثل العمر والمتوسط. لا يتم تخزين الخصائص المشتقة في الجدول، وإنما يتم حسابها عند الحاجة إليها .
4. الخصائص متعددة القيم (Multivalued Properties): هي الخصائص التي يمكن أن تحتوي على أكثر من قيمة، مثل اللغات المتحدث بها أو الهوايات المفضلة

المفاتيح الأساسية في مخطط الكائنات العلاقية :-

1. المفتاح الاساسى (Primary Key) :- هي الخصائص التي تحدد هوية الكائنات وتميزها عن الكائنات الأخرى. وتستخدم الخصائص الأساسية عادة كرابط بين الكائنات في العلاقات. ويتم تعيين قيمة فريدة لكل خاصية أساسية.

يتميز المفتاح الأساسي بأنه يحدد هوية الكائنات ويميزها عن الكائنات الأخرى، وهذا يساعد في تمييز الصفوف والوصول إلى البيانات بشكل أسرع وأكثر دقة. تستخدم المفاتيح الأساسية في مخطط الكائنات العلاقية لعدة أغراض، ومنها:

- ⇨ تحديد علاقات الجداول: حيث يمكن استخدام المفتاح الأساسي لربط الجداول ببعضها البعض عن طريق العلاقات.
- ⇨ ضمان عدم تكرار البيانات: حيث يتم تعيين قيمة فريدة لكل مفتاح أساسي، وهذا يضمن عدم تكرار البيانات في الجدول.
- ⇨ تحديد البيانات التي يتم البحث عنها: حيث يمكن استخدام المفتاح الأساسي للعثور على صف محدد في الجدول بشكل سريع ودقيق.
- ⇨ تحديد العمليات التي يمكن تطبيقها على الجدول: حيث يمكن استخدام المفتاح الأساسي لتحديد العمليات التي يمكن تطبيقها على الجدول، مثل الإدخال والتحديث والحذف.

2. المفتاح الاجنبى (Foreign Key) :- هي الخصائص التي تربط الكائنات ببعضها البعض من خلال العلاقات. وتحتوي الخصائص الأجنبية على قيم الخصائص الأساسية للكائنات الأخرى.

يساعد في ربط الجداول ببعضها البعض وتحديد العلاقات بينها. كما يساعد المفتاح الأجنبي في تحسين أداء قاعدة البيانات وجعلها أكثر فعالية، حيث يسهل البحث عن البيانات وحمايتها من الحذف أو التعديل غير المقصود. يتميز المفتاح الأجنبي بأنه يحتوي على قيم الخصائص الأساسية للكائنات في جدول آخر، ويتم استخدامه لربط الجداول ببعضها البعض.

تستخدم المفاتيح الأجنبية في مخطط الكائنات العلاقية لعدة أغراض، ومنها:

- ⇨ الحفاظ على السلامة والتسلسل: حيث يتم استخدام المفتاح الأجنبي لضمان أن العلاقة بين الجدولين سليمة ومنسجمة، وأن البيانات متسلسلة ومنظمة بشكل صحيح.
- ⇨ حماية البيانات: حيث يمكن استخدام المفتاح الأجنبي لحماية البيانات من الحذف أو التعديل غير المقصود من قبل المستخدمين، حيث يتم منع حذف أو تعديل الصفوف المرتبطة بالمفتاح الأجنبي.
- ⇨ البحث عن البيانات: حيث يمكن استخدام المفتاح الأجنبي للعثور على البيانات في جدول آخر، وذلك من خلال العلاقات المرتبطة بين الجدولين.
- ⇨ التحديث الآلي: حيث يمكن استخدام المفتاح الأجنبي لتحديث البيانات تلقائيًا، حيث يتم تحديث البيانات في جدول واحد بناءً على التغييرات التي تم إجراؤها في جدول آخر.

3. المفتاح الثانوي (Secondary Key) :- هو عنصر هام في قواعد البيانات، ويستخدم لتحسين أداء عمليات البحث والترتيب في جدول معين. ويمكن استخدام المفتاح الثانوي في العديد من الاستخدامات، بما في ذلك:

- ⇐ تسريع عمليات البحث: يمكن استخدام المفتاح الثانوي لتسريع عمليات البحث في جدول معين، حيث يتم إنشاء فهرس على القيم المحددة في المفتاح الثانوي، مما يسهل الوصول إلى الصفوف التي تحتوي على هذه القيم.
- ⇐ الترتيب والتصفية: يمكن استخدام المفتاح الثانوي لترتيب الصفوف في جدول معين، أو لتصفية الصفوف بناءً على القيم المحددة في المفتاح الثانوي.
- ⇐ الانضمام إلى جداول أخرى: يمكن استخدام المفتاح الثانوي للانضمام إلى جداول أخرى وربط الجداول ببعضها البعض، حيث يتم استخدام المفتاح الثانوي في الجدول الأول كمفتاح أجنبي في الجدول الثاني.
- ⇐ تحديد العلاقات بين الجداول: يمكن استخدام المفتاح الثانوي لتحديد العلاقات بين الجداول، حيث يتم استخدام قيم المفتاح الثانوي في الجدول الأول لتحديد الصفوف المرتبطة بقيم المفتاح الأجنبي في الجدول الثاني.
- ⇐ الاستعلامات المعقدة: يمكن استخدام المفتاح الثانوي في الاستعلامات المعقدة التي تتطلب البحث عن الصفوف بناءً على مجموعة من القيود والشروط المختلفة.

الفرق بين المفتاح الاساسى والمفتاح الثانوى والمفتاح الاجنبى :-

- ⇐ المفتاح الأساسي :- هو المفتاح الرئيسي في جدول البيانات، ويتميز بأنه يحتوي على قيم فريدة لكل صف في الجدول، ويتم استخدامه لتحديد الصفوف بشكل فريد وتمييزها عن بقية الصفوف. ويتم تحديد المفتاح الأساسي عادةً باختيار حقل يتميز بالفريدة والثبات، مثل رقم الهوية أو رقم الطلب أو رقم الحساب، ويتم استخدامه كمفتاح رئيسي للجدول.
 - ⇐ المفتاح الثانوي :- هو مفتاح يستخدم لتحسين أداء عمليات البحث والترتيب في جدول معين، ولا يعتبر مفتاحًا فريدًا للصفوف كما هو الحال في المفتاح الأساسي. ويتميز المفتاح الثانوي بأنه يتم إنشاؤه بناءً على الخصائص الأخرى في الجدول ولا يحتوي على قيم فريدة لكل صف في الجدول.
 - ⇐ المفتاح الأجنبي :- هو عبارة عن مفتاح يستخدم لربط جدول مع جدول آخر، ويحتوي على قيم تشير إلى الكائنات في الجدول الآخر. ويتم استخدام المفتاح الأجنبي لتحديد العلاقات بين الجداول وضمان سلامة البيانات في قاعدة البيانات.
- بشكل عام، يمكن القول إن المفتاح الأساسي يستخدم لتحديد الصفوف بشكل فريد وتمييزها عن بقية الصفوف، ويتم استخدامه كمفتاح رئيسي للجدول، بينما يستخدم المفتاح الثانوي لتحسين أداء البحث والترتيب في جدول معين، ويستخدم المفتاح الأجنبي لربط الجداول ببعضها البعض وتحديد العلاقات بينها.

تحويل نموذج العلاقة الكيانية (ERD) إلى مخطط الجدول (Schema)

مخطط قواعد البيانات : Schema Database هو مخطط يصف قاعدة البيانات بشكل رسومي تمهيدا لبنائه على شكل جداول في نظام إدارة قواعد بيانات DBMS

يتطلب مجموعة من الخطوات الأساسية التي يجب اتباعها بعناية للحصول على مخطط دقيق ومنظم. وفيما يلي الخطوات الأساسية لتحويل ERD إلى Schema بالتفصيل:

1. تحديد الكائنات (Entities) في ERD:

- ⇐ يجب تحديد الكائنات الموجودة في ERD، والتي تمثل الجداول في Schema.
- ⇐ يجب تحديد أسماء الكائنات بدقة ووضوح، والتأكد من أنها تعبر عن البيانات التي تحتويها كل جدول.
- ⇐ يجب تحديد الأعمدة (Columns) لكل جدول، وتحديد المفتاح الرئيسي (Primary Key) لكل جدول.

2. تحديد العلاقات (Relationships) بين الكائنات في ERD:

- ⇐ يجب تحديد العلاقات بين الكائنات في ERD، وتحويلها إلى علاقات بين الجداول في Schema.
- ⇐ يجب تحديد نوع العلاقة بين الجداول، مثل العلاقة الواحد إلى الواحد (One-to-One) أو العلاقة الواحد إلى العديد (One-to-Many) أو العلاقة العديد إلى العديد (Many-to-Many).
- 1. تحويل العلاقات الثنائية من النوع (1:1): إذا كانت العلاقة بين الكيانين علاقة واحد-إلى-واحد فإن عملية التحويل تتم وفق عدة خيارات أشهرها،
 - ⇐ خيار يسمى بطريقة المفتاح الأجنبي، وفيه يتم إضافة المفتاح الرئيسي لاجدولين إلى الجدول الآخر كمفتاح أجنبي ويفضل أن يكون الجدول الذي يحتوي على المفتاح الأجنبي، هو الجدول الذي يكون نوع قيد اشتراك في العلاقة، من نوع (الاشتراك الكلي)
 - ⇐ إذا كان الكيانان مرتبطان ارتباطا كليا بالعلاقة فيمكن اختيار أي كيان لضم المفتاح الأساسي للكيان الآخر إليه كمفتاح أجنبي
- 2. تحويل العلاقة الثنائية من النوع (N:1) يتم هنا إنشاء جدولين لتمثيل الكيانين المرتبطين، على أن يتم تطبيق طريقة المفتاح الأجنبي السابقة، وذلك بإضافة المفتاح الرئيسي للجدول من جهة العلاقة (1) إلى الجدول الآخر المرتبط بالعلاقة (N)، بغض النظر عن نوع قيد الاشتراك. ونضيف أية صفات موجودة على العلاقة إلى الجدول الآخر المرتبط بالعلاقة (N).
- 3. تحويل العلاقة الثنائية من النوع (M:N) في هذا النوع من العلاقات، يتم استحداث جدول جديد، فيكون الناتج من هذه العلاقة ثلاثة جداول، جدولين لتمثيل الكيانين المرتبطين بالعلاقة ويضم الجدول الثالث حقلين كمفتاحين أجنبيين يمثلان المفتاحين الرئيسيين في الجدولين، ويمكن إضافة أي حقل آخر يكون له مغزى، كأن تكون العلاقة لها صفة بذاتها، فتتحوّل الصفة إلى حقل في الجدول الجديد. المفتاح الأساسي للجدول الجديد هو مجموعة المفاتيح الأجنبية التي تم ضمها إليه وتمثل المفاتيح الأساسية للكيانين المرتبطين بالعلاقة.
- ⇐ يجب تحديد المفاتيح الأجنبية (Foreign Keys) لكل جدول، والتي تربط الجدول بعلاقة محددة.

3. تحديد القيود (Constraints) الأخرى:

- ⇐ يجب تحديد القيود الفريدة (Unique Constraints) والقيود الداخلية (Check Constraints) والقيود الخارجية (Referential Integrity Constraints) وتطبيقها على الجداول في Schema.

← يجب تحديد المفاتيح الفرعية (Secondary Keys) وتطبيقها على الجداول في Schema.

4. تحويل الخصائص:

← تحويل الصفات متعددة القيم: يتم في هذه الحالة، عادة، إنشاء جدول جديد يضم الصفة المتعددة القيم كحقل، ويضاف إلى الجدول مفتاح أجنبي FK يكون ممثلاً للمفتاح الرئيسي في الجدول الناتج من الكيان الذي يحتوي على الصفة متعددة القيم .

← أما الصفات المركبة فتتحوّل إلى صفات بسيط، فحقول عادية

← الصفات ذات القيم المشتقة تلغى من الجدول، لأنها صفات قابلها لالشتقاق من صفات أخرى، فلا داعي لوجودها

الفرق بين مخطط الكائنات العلائقية وال Schema :-

ERD هي اختصار Entity Relationship Diagram وتشير إلى نموذج بيانات يستخدم لوصف العلاقات بين الكيانات في قاعدة البيانات. وبمعنى آخر، فإن ERD يعرض بيانات الكيانات والعلاقات بينها في شكل رسم بياني.

Schema فهي تصميم يستخدم لوصف بنية البيانات في قاعدة البيانات. ويحدد schema الأنواع والصيغ والعلاقات بين البيانات المختلفة الموجودة في قاعدة البيانات.

يمكن القول أن ERD هو نوع من ال schema يركز بشكل خاص على تمثيل العلاقات بين الكيانات في قاعدة البيانات، ويستخدم الرسوم البيانية لتوضيح هذه العلاقات. ومن خلال ERD يمكن توضيح العلاقات الوظيفية بين الكيانات والمفاهيم المختلفة في قاعدة البيانات. ويمكن استخدام ERD كأداة لتطوير schema. ويتم استخدام schema لتحديد البيانات المختلفة الموجودة في قاعدة البيانات والعلاقات بينها، بما في ذلك العلاقات بين الجداول والحقول والمفاتيح الخارجية وغيرها.

Normalizing a database :-

عملية تطبيع قاعدة البيانات تعني تحويل الجداول غير المنتظمة إلى جداول منتظمة ومنظمة بطريقة تسمح بتحليلها بسهولة ومرونة. وتتم هذه العملية من خلال تقسيم الجدول الأساسي إلى جداول فرعية مترابطة، وتجنب التكرار في المعلومات، وتجنب إدخال بيانات متناقضة أو متضاربة.

يتم تصميم قاعدة البيانات بحيث تتكون من مجموعة من الجداول، وتتمثل العلاقة بين الجداول في استخدام مفتاح خارجي في جدول واحد للربط بينه وبين مفتاح رئيسي في جدول آخر. ويجب أن تكون هذه العلاقات مدروسة بعناية للحصول على تصميم قاعدة بيانات فعال وفعال.

يتم تحويل الجدول الأساسي إلى جداول فرعية عن طريق تطبيق مجموعة من قواعد التطبيع والتي تنقسم إلى الأشكال التالية:

1. الشكل الأول (1NF - First Normal Form):-

للوصول إلى المستوى الأول في مخطط الكائنات العلائقية NF1، يجب تصميم قاعدة البيانات بحيث تستوفي المتطلبات التالية:

← كل جدول في قاعدة البيانات يجب أن يحتوي على خلايا واحدة فقط لكل القيم.

- ⇐ لا يجب أن تحتوي الخلايا في الجدول على مجموعات من القيم أو أي قيم متعددة.
- ⇐ يجب تحديد مفتاح رئيسي فريد لكل صف في الجدول.
- ⇐ يجب أن تكون الأعمدة في الجدول متجانسة من حيث النوع والتنسيق.
- ⇐ يجب أن يكون لكل جدول في قاعدة البيانات اسم فريد وصحيح.
- ⇐ يجب أن يكون الجداول مستقلة عن بعضها البعض، ولا يجب أن يكون هناك أي تداخل في البيانات بين الجداول.

لتحقيق هذه المتطلبات، يجب تصميم قاعدة البيانات بعناية وتجنب تكرار البيانات وتجزئة البيانات بين الجداول. يجب أيضًا تحديد مفتاح رئيسي فريد لكل جدول وتجنب استخدام القيم المتعددة في الخلايا.

2. الشكل الثاني (Second Normal Form - 2NF):

للوصول إلى المستوى الثاني في مخطط الكائنات العلائقية NF2، يجب تصميم قاعدة البيانات بحيث تستوفي المتطلبات التالية:

- ⇐ يجب أن يكون الجدول في المستوى الأول في NF1.
 - ⇐ يجب تحديد المفتاح الرئيسي للجدول وتأكيد أن كل الأعمدة في الجدول تعتمد بشكل كامل على المفتاح الرئيسي.
 - ⇐ يجب فصل البيانات غير المرتبطة بالمفتاح الرئيسي إلى جدول منفصل.
 - ⇐ يجب تحديد المفتاح الرئيسي للجدول الجديد وتأكيد أن كل الأعمدة في الجدول الجديد تعتمد بشكل كامل على المفتاح الرئيسي الجديد.
- بمعنى آخر، يجب تفكيك الجدول إلى جداول أصغر وأكثر تفصيلاً، حيث يحتوي كل جدول على بيانات تعتمد بشكل كامل على مفتاح رئيسي واحد. يساعد هذا على تقليل تكرار البيانات وزيادة الكفاءة في قاعدة البيانات.

3. الشكل الثالث (Third Normal Form - 3NF):

للوصول إلى المستوى الثالث في مخطط الكائنات العلائقية NF3، يجب تصميم قاعدة البيانات بحيث تستوفي المتطلبات التالية:

- ⇐ يجب أن يكون الجدول في المستوى الثاني في NF2.
- ⇐ يجب تأكيد أن كل الأعمدة في الجدول تعتمد بشكل كامل على المفتاح الرئيسي، ولا تحتوي على أي بيانات متداخلة أو غير ضرورية.
- ⇐ يجب فصل البيانات غير المرتبطة بالمفتاح الرئيسي إلى جدول منفصل.

يتعلق المستوى الثالث من مخطط الكائنات العلائقية بتصميم البيانات بحيث يتم تقليل التكرار وتجزئة البيانات بشكل فعال. وبالتالي، يجب تفكيك الجدول إلى جداول أصغر وأكثر تفصيلاً، حيث يحتوي كل جدول على بيانات تعتمد بشكل كامل على مفتاح رئيسي واحد، ولا تحتوي على أي بيانات متداخلة أو غير ضرورية.

4. الشكل الرابع (Fourth Normal Form - 4NF):

للوصول إلى المستوى الرابع في مخطط الكائنات العلائقية NF4، يجب تصميم قاعدة البيانات بحيث تستوفي المتطلبات التالية:

- ⇐ يجب أن يكون الجدول في المستوى الثالث في NF3.
- ⇐ يجب تجزئة البيانات غير المرتبطة بالمفتاح الرئيسي إلى جداول منفصلة، ولكن يجب تجنب تكرار البيانات بين الجداول.
- ⇐ يجب تحديد العلاقات بين الجداول باستخدام المفاتيح الأجنبية (Foreign Keys)، وضمان عدم وجود بيانات متداخلة بين الجداول.
- ⇐ يجب تحديد المفاتيح الفرعية (Secondary Keys) لجدول العلاقات العديدة إلى العديد (-Many to-Many)، والتي تتضمن جداول وسيطة لتمكين العلاقة بين الجداول.
- بمعنى آخر، يجب تجزئة الجداول إلى جداول أصغر وأكثر تفصيلاً، وتحديد العلاقات بين الجداول باستخدام المفاتيح الأجنبية، وتحديد المفاتيح الفرعية لجدول العلاقات العديدة إلى العديد.
- يمكن استخدام العلاقات العديدة إلى العديد (Many-to-Many) لحل المشكلات التي تواجه الجداول التي تحتوي على بيانات متعددة القيم، والتي يجب تجزئتها إلى جداول منفصلة، ولكن يمكن استخدام جداول وسيطة لربط الجداول المختلفة.

5. الشكل الخامس (Fifth Normal Form - 5NF):

للوصول إلى المستوى الخامس في مخطط الكائنات العلائقية NF5، يجب تصميم قاعدة البيانات بحيث تستوفي المتطلبات التالية:

- ⇐ يجب أن يكون الجدول في المستوى الرابع في NF4.
- ⇐ يجب تجزئة البيانات بحيث يتم تفكيك الجدول إلى مجموعة من الجداول الصغيرة المرتبطة بشكل معقول.
- ⇐ يجب تحديد العلاقات بين الجداول باستخدام المفاتيح الأجنبية (Foreign Keys)، وضمان عدم وجود بيانات متداخلة بين الجداول.
- ⇐ يجب تحديد المفاتيح الفرعية (Secondary Keys) لجدول العلاقات العديدة إلى العديد (-Many to-Many)، والتي تتضمن جداول وسيطة لتمكين العلاقة بين الجداول.
- ⇐ يجب تحديد الاعتمادات الوظيفية (Functional Dependencies) بين الأعمدة في الجدول، وضمان عدم وجود اعتمادات وظيفية زائدة.
- بمعنى آخر، يجب تفكيك الجدول إلى جداول أصغر وأكثر تفصيلاً، وتحديد العلاقات بين الجداول باستخدام المفاتيح الأجنبية، وتحديد المفاتيح الفرعية لجدول العلاقات العديدة إلى العديد، وتحديد الاعتمادات الوظيفية بين الأعمدة في الجداول.
- تستخدم عملية تطبيع قاعدة البيانات لتحسين أداء قاعدة البيانات وجعلها أكثر كفاءة ومرونة، كما تساعد على تحسين تنظيم البيانات وتقليل تكرارها، مما يؤدي إلى توفير مساحة تخزينية وتسهيل عمليات البحث والتحديث في البيانات. كما أنها تساعد في تحقيق أهداف الانضباط والدقة في البيانات، حيث يتم تجنب إدخال بيانات متناقضة أو متضاربة.

ومن بين استخدامات تطبيع قاعدة البيانات:

1. تحسين أداء قاعدة البيانات: حيث يتم تخفيض حجم البيانات وتحسين تنظيمها، مما يؤدي إلى تحسين أداء عمليات البحث والتحديث في البيانات.
2. تحسين دقة البيانات: حيث يتم تجنب إدخال بيانات متناقضة أو متضاربة، وتحسين جودة البيانات المخزنة.
3. توفير مساحة تخزينية: حيث يتم تحسين تنظيم البيانات وتجنب تكرارها، مما يؤدي إلى توفير مساحة تخزينية.
4. تسهيل الصيانة: حيث يتم تحسين تنظيم البيانات وتجنب التكرار، مما يسهل عمليات الصيانة والتحديث في قاعدة البيانات.
5. توفير المرونة: حيث يتم تحسين تنظيم البيانات وجعلها أكثر مرونة وسهولة في التعديل والتحديث.

مخطط السياق (Context Diagram)

- ← هو عبارة عن مخطط بياني يوضح العلاقة بين نظام معين وبيئته الخارجية.
- ← يعرض المخطط الكيفية التي يتفاعل بها النظام مع العوامل الخارجية المحيطة به.
- ← يتم استخدام مخطط السياق في تحليل النظام وتصميمه ووصفه .

تستخدم مخطط السياق في العديد من الأغراض، منها:

- ← توضيح وصف عام لنظام معين والعلاقات التي يتفاعل بها مع بيئته الخارجية.
- ← توضيح المدخلات والمخرجات الرئيسية للنظام وكيفية تدفق المعلومات من وإلى النظام.
- ← توضيح الكيانات الرئيسية المتفاعلة مع النظام وعلاقاتها به.
- ← توضيح الأنظمة الفرعية التي تتفاعل مع النظام الرئيسي وكيفية تدفق المعلومات بينهما.

من المميزات الرئيسية لمخطط السياق:

- ← يوضح بشكل واضح العلاقات بين النظام والعوامل الخارجية المحيطة به.
- ← يمكن استخدامه في جميع مراحل تصميم النظام، بدءًا من التحليل وحتى التنفيذ والاختبار.
- ← يعتبر مخططًا بسيطًا وسهل الفهم ويمكن تطبيقه بسرعة.

تتكون مخطط السياق من العناصر التالية:

- ← النظام: هو المركز الرئيسي في المخطط والذي يتم تحليله وتصميمه.
- ← المدخلات: هي المعلومات التي يحتاجها النظام للعمل بشكل صحيح.
- ← المخرجات: هي المعلومات التي ينتجها النظام بعد معالجة المدخلات.
- ← الكيانات الخارجية: هي الكيانات التي تتفاعل مع النظام وتؤثر عليه.

مخطط تدفق البيانات (Data Flow Diagram)

- ← هو أداة رسومية تستخدم لتوضيح كيفية تدفق المعلومات داخل نظام معين.

← يستخدم المخطط لتمثيل المعالجات المختلفة التي يتم تطبيقها على المعلومات، والمخازن التي يتم تخزين المعلومات فيها، والمدخلات والمخرجات الرئيسية للنظام.

يتم استخدام مخطط تدفق البيانات في العديد من الأغراض، بما في ذلك:

- 1- توضيح تدفق المعلومات داخل النظام وتحليله .
- 2- تصميم النظام وتطويره: يمكن استخدام المخطط لتصميم النظام وتطويره، حيث يمكن استخدامه لتحديد المكونات المختلفة للنظام وعلاقاتها.
- 3- تحسين النظام: يمكن استخدام المخطط لتحسين النظام وتحديد المناطق التي يمكن تحسينها وتطويرها.
- 4- توثيق النظام: يمكن استخدام المخطط لتوثيق النظام وتوضيح كيفية عمله وتفاعل مكوناته.

من مميزات مخطط تدفق البيانات:

- 1- سهولة التفاعل مع المعلومات: يساعد المخطط في توضيح تفاصيل تدفق المعلومات داخل النظام وتسهيل التفاعل معها.
- 2- التحليل والتصميم الفعال: يمكن استخدام المخطط لتحليل وتصميم النظام بشكل فعال وتحديد المكونات المختلفة للنظام وعلاقاتها.
- 3- الوضوح والدقة: يساعد المخطط في توضيح تفاصيل تدفق المعلومات بشكل واضح ودقيق.

يتكون مخطط تدفق البيانات من العديد من العناصر، بما في ذلك:

- 1- المدخلات (Inputs): وهي المعلومات التي تدخل إلى النظام من العوامل الخارجية.
- 2- المعالجات (Processes): وهي العمليات المختلفة التي يتم تطبيقها على المعلومات داخل النظام لتحويلها إلى معلومات أخرى.
- 3- المخرجات (Outputs): وهي المعلومات التي يتم إنتاجها من خلال تطبيق المعالجات على المدخلات.

- 4- المخازن (Data Stores): وهي الأماكن التي يتم تخزين المعلومات فيها داخل النظام.
- 5- الربط (Data Flows): وهي الروابط التي تربط بين المدخلات والمعالجات والمخرجات والمخازن، وتوضح كيفية تدفق المعلومات داخل النظام.
- 6- الحدود (Boundaries): وهي الخطوط التي تفصل النظام عن العوامل الخارجية، وتوضح الحدود بين النظام والبيئة التي يعمل فيها.

بشكل عام، يستخدم مخطط تدفق البيانات لتوضيح كيفية تدفق المعلومات داخل النظام وتحليله، ويمكن استخدامه في تصميم وتطوير النظام وتحسينه وتوثيقه وتسهيل التواصل بين فرق التطوير والمستخدمين.

خصائص مخطط تدفق البيانات

إن لتصميم مخططات تدفق البيانات شروطاً كثيرةً يجب التقيد بها حتى يصبح التواصل من خلالها واضحاً، ومن أهم الشروط ما يلي:

- ⇐ ألا تحتوي على تدفقات بيانات بين المصادر الخارجية مع بعضها البعض، حيث إن العلاقة بين المصادر بعضها ببعض لا تهم النظام.
- ⇐ ألا تحتوي على إشارات تحكم =، <، >.
- ⇐ ألا تحتوي على تكرار أو دورات.
- ⇐ ألا تحتوي على تدفقات بين المصادر الخارجية و مخازن البيانات مباشرة
- ⇐ التقيد بمبدأ "حفظ البيانات"، هذا يعني أن البيانات لا تكون موجودة في مخازن البيانات من دون أن تكون متدفقة من وظيفة ما و لا يمكن للبيانات أن تدخل النظام وتخرج منه إلى المصادر الخارجية أو تستقر في مخازن البيانات إلا عن طريق وظيفة ما.
- ⇐ يجب الحرص على تسمية كافة عناصر المخطط.
- ⇐ يجب الحرص على الترقيم الواضح للمخطط العام والمخططات التفصيلية.
- ⇐ التأكد من أن التدفقات الداخلة إلى المخطط العام هي نفسها الداخلة إلى المخططات التفصيلية وكذلك الحرص على وجود مخازن البيانات
- ⇐ يمكن تكرار المخازن أو المصادر لتوضيح الرسم وتنظيمه .
- ⇐ يلاحظ أن الوظائف الرئيسية في نموذج الوظائف هي نفسها المستوى الصفري (العام) لمخطط تدفق البيانات، بينما تفرع كل وظيفة من وظائف مخطط الوظائف هي المخططات التفصيلية لمخطط تدفق البيانات.

عناصر مخطط تدفق البيانات :-

1- مخزن البيانات (Data Store):

- ⇐ يستخدم مخزن البيانات لتمثيل مخازن البيانات الموجودة في النظام وهي الملفات أو جداول قواعد البيانات (يسمى مخزن معلومات داخلي) .
- ⇐ يتم تمثيل مخزن البيانات بشكل مستطيل مفتوح من احد الجانبين
- ⇐ يتم تسمية مخازن البيانات باسم جمع يدل على عدد السجلات الموجودة
- ⇐ لا يتم تحويل البيانات أو تغييرها داخل مخازن البيانات، فالبيانات المدخلة هي نفسها المخرجة ولا تظهر في المخطط البيئي لكونها عناصر داخلية للنظام، ويمكن تكرار رسمها في المخطط بهدف تسهيل وتنظيم الرسم .

2- العملية (Process)

- ⇐ تعتبر المكون الرئيسي لمخطط تدفق البيانات.
- ⇐ هي أي عمل يتم انجازه لتحويل البيانات الداخلة إلى بيانات خارجة.
- ⇐ اسم الوظيفة يكتب داخل رمز العملية وهو عبارة عن فعل أمر يدل على معناه بوضوح ويرمز لها بشكل دائري يكتب فيه اسم الوظيفة ورقمها.
- ⇐ لا يتم تحديد كيفية أداء الوظيفة أو كيف تمت المعالجة.
- ⇐ يتم التركيز هنا على البيانات المدخلة والبيانات المخرجة التي تكون قد اختلفت نتيجة عمل معين أو معالجة معينة عليها.

3- تدفق البيانات (Data Flow):

- ⇐ يستخدم لتوضيح عملية تدفق البيانات واتجاهها داخل النظام بين الوظائف، وكذلك لتوضيح تدفق البيانات بين المصادر الخارجية إلى النظام والعكس.
- ⇐ يرمز لها بسهم ويُسمى حسب نوع البيانات ولا يوضح في المخطط درجة تكرار البيانات.

4- المصادر الخارجية (External Resources):

← يستخدم لتمثيل المصادر الخارجية للنظام، أي عناصر البيئة الخارجية للنظام التي تتبادل معه البيانات.

← يرمز لها بشكل مستطيل يكتب داخله اسم الجهة.

لماذا لا تظهر مخازن البيانات في مخطط السياق

يتم استخدام مخطط السياق (Context Diagram) لتوضيح العلاقات بين النظام والعوامل الخارجية، ويركز على المدخلات والمخرجات الرئيسية للنظام والكيانات المتفاعلة معه. وبما أن مخازن البيانات لا تعد كيانات متفاعلة مع النظام في حد ذاتها، فإنها غالبًا ما لا تظهر في مخطط السياق.

أيه الفرق بين الـ DFD و الـ ERD

DFD هي اختصار Data Flow Diagram وتشير إلى نموذج بيانات يستخدم لوصف تدفق البيانات والعمليات في نظام معين. وبمعنى آخر، فإن DFD يعرض المعلومات التي تتحرك بين العمليات والأنظمة المختلفة في شكل رسم بياني.

أما ERD فهي اختصار Entity Relationship Diagram وتشير إلى نموذج بيانات يستخدم لوصف العلاقات بين الكيانات في قاعدة البيانات. وبمعنى آخر، فإن ERD يعرض بيانات الكيانات والعلاقات بينها في شكل رسم بياني.

يمكن القول إن الفرق الرئيسي بين DFD و ERD هو أن DFD يركز بشكل خاص على توضيح تدفق البيانات والعمليات التي تحدث في النظام، بينما يركز ERD بشكل خاص على توضيح العلاقات بين الكيانات في قاعدة البيانات.

وعلى الرغم من أنهما يستخدمان رسومات بيانية لتوضيح البيانات والعلاقات، فإن استخدام DFD و ERD يختلف تمامًا. حيث يتم استخدام DFD لتصميم وتطوير نظام المعلومات، بينما يتم استخدام ERD لتصميم وتطوير قاعدة البيانات.

ويجب الإشارة إلى أنه على الرغم من أن DFD و ERD مختلفان، إلا أنهما يمكن أن يستخدمان معًا في بعض الأحيان لتصميم وتطوير نظام المعلومات الذي يعتمد على قاعدة البيانات. فعلى سبيل المثال، يمكن استخدام DFD لتحديد تدفق البيانات في النظام، واستخدام ERD لتحديد الكيانات والعلاقات بينها في قاعدة البيانات.

UML هي اختصار لـ "Unified Modeling Language"

هي لغة نمذجة مستخدمة في تصميم ووصف نماذج البرمجيات ونماذج الأنظمة المعقدة.

← تستخدم UML في تحليل وتصميم البرمجيات، وتوضيح تصميم الكائنات والصفوف والعلاقات بينها.

← توثيق النظم وتحديد المتطلبات البرمجية

← تصميم ووصف النظم المعقدة والأنظمة الموزعة.

أنواع مخططات UML

تستخدم لتصميم وتخطيط المشاريع البرمجية قبل البدء العمل فيها وتستخدم مجموعة من الرموز والاشكال الهندسية كل رمزه يدل على شئ معين في النظام.

أهم خصائصها :-

- ⇐ ليست منهجية لتصميم وبناء البرمجيات وتطويرها . (منهجية زي نموذج الشلال كذا) .
 - ⇐ لا ترتبط بلغة برمجة أو بمنهجية محددة لبناء البرمجيات .
 - ⇐ يتم استخدامها في مرحلة التصميم .
 - ⇐ تقوم بتوصيف عمليات النظام فقط وليس لها علاقة بكيفية برمجة وظائف النظام .
- هناك عدة أنواع من مخططات UML التي تستخدم لتصميم ووصف النظم والبرمجيات، ويوجد 14 نوعًا من المخططات يتم تقسيمها بشكل عام الى ثلاث أنواع رئيسية ومن بين هذه المخططات:

1- مخططات التصميم الهيكلية (Structural Diagrams):

هي المخططات التي تستخدم لوصف وتوضيح هيكل النظام والعلاقات بين مكوناته

- ⇐ مخططات الصفوف (Class Diagrams)
- ⇐ مخططات الكائنات (Object Diagrams)
- ⇐ مخططات الحزم (Package Diagrams)
- ⇐ مخططات الترتيب (Sequence Diagrams)
- ⇐ مخططات التعاون (Collaboration Diagrams)
- ⇐ مخططات الحالة (State Diagrams).

2- مخططات التصميم السلوكي (Behavioral Diagrams):

هي المخططات التي تستخدم لوصف سلوك النظام وتفاعلاته مع المستخدمين والعناصر الأخرى،

- ⇐ مخططات الأنشطة (Activity Diagrams)
- ⇐ مخططات التوقيت (Timing Diagrams)
- ⇐ التفاعل (Interaction Overview Diagrams).
- ⇐ مخطط حالات الاستخدام (Use Case Diagram)

3- مخططات التصميم العملي (Implementation Diagrams):

هي المخططات التي تستخدم لوصف تنفيذ النظام وتحديد المكونات الفعلية التي يتم استخدامها في تطوير النظام

- ⇐ مخططات التثبيت (Deployment Diagrams)
- ⇐ مخططات التكوين (Component Diagrams).

يمكن استخدام هذه المخططات بشكل منفصل أو مجتمعين لتوضيح جوانب مختلفة من تصميم النظام ووصفه.

هستخدام منهم 4 مخططات

1. مخطط حالات الاستخدام (Use Case Diagram)

- ← هو أحد مخططات UML ويستخدم لوصف الممارسات والأنشطة التي يقوم بها المستخدمون أو الأنظمة في سياق معين.
- ← يعرض المخطط الممثلين الرئيسيين للنظام والممارسات التي يمكن أن يقوموا بها وكيف يرتبطون بالنظام.
- ← مخطط يستخدم لعرض العلاقة بين الجهات الفاعلة وحالات الاستخدام .

يتكون المخطط من عناصر رئيسية هي الممثلين (Actors) وحالات الاستخدام (Use Cases) والعلاقات بينهما.

يتم تمثيل الممثلين بأي عنصر يتفاعل مع النظام، مثل المستخدمين أو الأجهزة الخارجية، يتم تمثيل حالات الاستخدام بأي نشاط يمكن أن يتم تنفيذه في النظام، مثل إجراءات البحث أو إضافة معلومات جديدة.

يستخدم مخطط حالات الاستخدام في مرحلة تحليل وتصميم النظام، ويمكن استخدامه أيضاً في توضيح العلاقات بين الممثلين وحالات الاستخدام للمستخدمين والمطورين والمسؤولين.

أنواع من العلاقات التي يمكن تعريفها بين حالات الاستخدام في مخطط حالات الاستخدام Use Case Diagram

1- العلاقة الأساسية (Basic Relationship): هي العلاقة الرئيسية بين الممثلين وحالات الاستخدام، وتشير إلى أن الممثل يشارك في تفاعل مع النظام من خلال تنفيذ حالات الاستخدام المعينة.

2- العلاقة الشاملة (Include Relationship): تشير إلى أن حالة الاستخدام الواحدة تتضمن حالة الاستخدام الأخرى، ويتم استخدام هذه العلاقة عندما يحتاج تنفيذ حالة الاستخدام الأساسية إلى تنفيذ حالة الاستخدام الأخرى.

3- العلاقة الاختيارية (Extend Relationship): تشير إلى أن حالة الاستخدام الواحدة يمكن تمديدها من خلال تنفيذ حالة الاستخدام الأخرى، ويتم استخدام هذه العلاقة عندما يحتاج تنفيذ حالة الاستخدام الأساسية إلى تنفيذ حالة الاستخدام الأخرى بشكل اختياري.

مكونات مخطط حالات الاستخدام:-



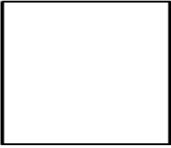
Actor

← الجهات الفاعلة Actors:- الفاعل في مخطط حالة الاستخدام هو أي كيان يؤدي دوراً في نظام معين، ويمكن أن يكون هذا شخصاً أو منظمة أو نظاماً خارجياً، وعادة ما يتم رسمه مثل الهيكل العظمي

← حالات الاستخدام Use Case:- تمثل حالة الاستخدام وظيفة أو إجراء داخل النظام، ويتم رسمه على شكل بيضاوي ويتم تسميته بالوظيفة التي يقوم بها .



Use Case



← النظام System :- يُستخدم النظام لتحديد نطاق رسم مخطط حالة الاستخدام ويتم رسمه على شكل مستطيل،

2. مخطط التتابع sequence

← مخطط التتابع (Sequence Diagram) هو أحد مخططات UML ويستخدم لوصف تفاعلات الكائنات والعمليات في النظام على مدار فترة زمنية محددة.

← يعرض المخطط التواصل بين العناصر المختلفة في النظام وتوقيت حدوث الأحداث.

مكونات مخطط التتابع

← يتكون المخطط من عناصر رئيسية هي الكائنات (Objects) والرسائل (Messages) والحدث (Events) والأزواج (Lifelines).

← يتم تمثيل الكائنات بأي عنصر في النظام يتفاعل مع كائنات أخرى، مثل الكائنات البرمجية أو الأجهزة الخارجية.

← يتم تمثيل الرسائل بأي نوع من الاتصالات بين الكائنات، مثل إرسال واستقبال البيانات.

أستخدام مخطط التتابع

← تستخدم الأحداث لتوضيح وقت حدوث الأحداث في المخطط، مثل إنشاء كائن جديد أو إرسال رسالة. وتمثل الأزواج الحد الزمني لتفاعل الكائنات مع بعضها البعض.

← يساعد مخطط التتابع في فهم وتوثيق تفاعلات النظام وتحليلها بشكل مفصل وواضح، تحديد تسلسل الأحداث وتوقيت حدوثها.

← يمكن استخدامه لتصميم واختبار البرمجيات وتبسيط عملية التطوير والصيانة.

← يستخدم مخطط التتابع في مرحلة تحليل وتصميم النظام، ويمكن استخدامه أيضاً في توضيح التفاعلات البرمجية للمطورين والمستخدمين والمسؤولين.

سبب وجود الفاصل " الخط المتقطع " بين كل رساله والتي تليها

الفاصل الرأسي في مخطط التتابع (Sequence Diagram) لا يعبر بشكل مباشر عن الوقت المستغرق بين كل عملية والتي تليها. بدلاً من ذلك، يعبر الفاصل الرأسي عن العلاقة الزمنية بين الرسائل والأحداث المختلفة في المخطط.

يمكن أن يعبر الفاصل الرأسي عن عدة نقاط في المخطط، مثل:

← التباعد الزمني بين رسائل الطلب والاستجابة.

← الفترة الزمنية التي تستغرقها عملية معينة.

← تباعد زمني بين بداية الحدث ونهايته.

ومع ذلك، يتم تحديد الوقت المستغرق بين كل عملية والتي تليها بشكل أكثر دقة باستخدام أدوات توقيت أخرى مثل مخططات التوقيت أو مخططات النشاط الزمنية.

أنواع الرسائل في مخطط التتابع

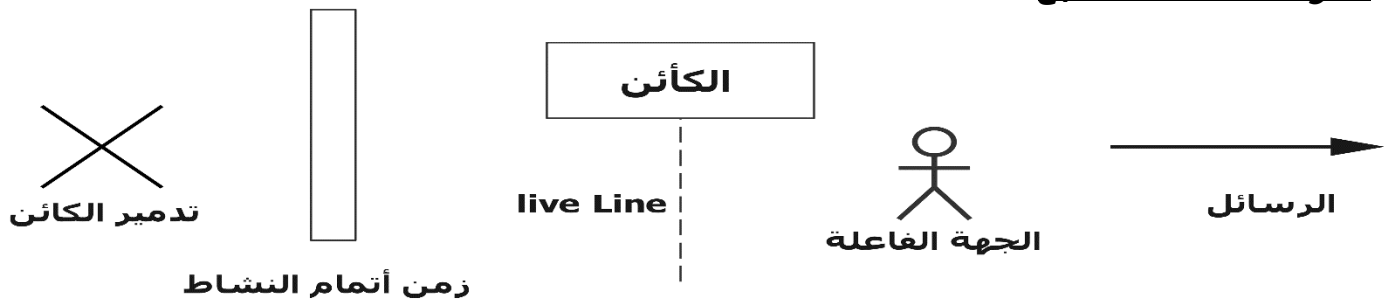
يوجد ثلاثة أنواع رئيسية من الرسائل في مخطط التتابع (Sequence Diagram):

1- رسالة الطلب (Request Message): هي الرسالة التي يقوم العنصر المرسل بإرسالها إلى العنصر المستقبل لطلب خدمة أو عملية محددة.

2- رسالة الاستجابة (Response Message): هي الرسالة التي يرسلها العنصر المستقبل بعد استلام رسالة الطلب، وتحتوي على النتيجة أو الاستجابة للطلب الذي تم إرساله.

3- رسالة الاستدعاء (Invocation Message): هي الرسالة التي يرسلها العنصر المستقل لدعوة العنصر الآخر لتنفيذ عملية محددة، وتختلف عن رسالة الطلب في أنها تحتوي على اسم العملية المطلوب تنفيذها، بدلاً من طلب خدمة أو عملية محددة ولا تحتاج لرد.

مكونات مخطط التتابع



أنواع الرسائل

الرسالة	المدلول
←	رسالة عادية ليس لها رد
←	رسالة استدعاء تحتاج لرد
- - - - - ➔	رد

3. مخطط النشاط Activity

⇐ مخطط النشاط (Activity Diagram) هو مخطط يستخدم لتصميم ووصف وتوضيح سير العمل أو العملية المختلفة في نظام معين.

⇐ يتمثل الهدف الرئيسي لمخطط النشاط في توضيح تسلسل الأحداث والخطوات المختلفة التي يجب اتخاذها لإكمال عملية محددة.

⇐ يتم تمثيل سير العمل في مخطط النشاط على شكل رسومات بيانية تتكون من عدة عناصر رئيسية، مثل:

1- النشاط (Activity): يمثل عملية محددة أو خطوة في السير العمل.

2- القرار (Decision): يمثل نقطة التحول في السير العمل، حيث يتم اتخاذ قرار بناءً على شرط محدد.

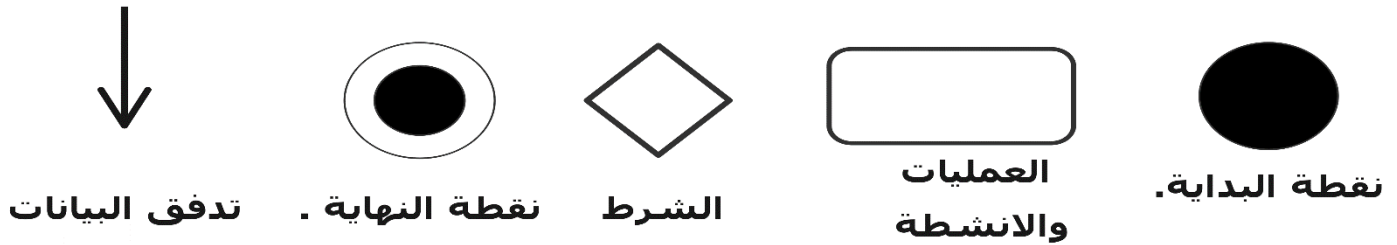
- 3- الانتظار (Wait): يمثل نقطة في السير العمل يتم فيها الانتظار لحدوث حدث محدد قبل المتابعة إلى الخطوة التالية.
- 4- الاتصال (Communication): يمثل تبادل المعلومات أو الرسائل بين الأطراف المختلفة في السير العمل.

ويتم ربط هذه العناصر بشكل منطقي باستخدام الروابط والسهام لتوضيح تسلسل الأحداث والخطوات في السير العمل.

أستخدام مخطط النشاط

- ⇐ يعد مخطط النشاط أداة هامة في تحليل العمليات وتصميمها وتوضيحها، حيث يساعد على فهم السير العمل بشكل أفضل وتحديد العمليات غير الفعالة وتحسينها.
- ⇐ مخطط يستخدم لوصف مسار التطبيق من نقطة البداية وحتى النهاية والأنشطة والعمليات التي تمر بها بشكل تسلسلي.

مكونات مخطط النشاطات



الفرق بين مخطط النشاط ومخطط التتابع

يوجد العديد من الاختلافات بين مخطط النشاط (Activity Diagram) ومخطط التتابع (Sequence Diagram)، ومن بين هذه الاختلافات:

- 1- الغرض: يستخدم مخطط النشاط لوصف سير العمل أو العملية المختلفة في نظام معين، بينما يستخدم مخطط التتابع لتمثيل تفاعلات العناصر المختلفة في النظام وتتبع تسلسل الرسائل بينها.
- 2- العناصر: يتكون مخطط النشاط من مكونات مختلفة مثل النشاط والقرار والحلقة والاتصال، بينما يتكون مخطط التتابع من عناصر مثل الكائنات والمسارات والرسائل.
- 3- الزمن: يتمثل الزمن في مخطط التتابع عن طريق ترتيب الرسائل في الزمن وتوقيت استجابة العناصر المختلفة، بينما لا يوجد تركيز كبير على الزمن في مخطط النشاط.
- 4- التركيز على الرسالة: يتركز مخطط التتابع على تتبع الرسائل وتفاصيل الاتصال بين العناصر، بينما يتركز مخطط النشاط على وصف الخطوات والعمليات اللازمة لإنجاز المهمة.

- 5- الدقة: يتطلب مخطط التتابع تفاصيل دقيقة حول ترتيب الرسائل وتوقيتها، بينما يمكن أن يكون مخطط النشاط أكثر قدرة على التعمق في التفاصيل وتوضيح الخطوات المحددة في العملية.
- 6- المستوى: يمكن استخدام مخطط التتابع على مستوى العمليات الفردية داخل النظام، بينما يمكن استخدام مخطط النشاط على مستوى أعلى لوصف سير العمل أو العملية بشكل عام.

4. مخطط الفئات Class

- ← مخطط الفئات (Class Diagram) هو مخطط بياني يستخدم في تصميم البرمجيات لتوضيح العلاقات بين الكائنات المختلفة في النظام وتوصيف هيكل النظام
- ← يمكن تعريف الفئة كمجموعة من الكائنات التي لها خواص مشتركة وسلوك مشترك.
- ← يتم تمثيل كل فئة في مخطط الفئات بصورة مربع يتم كتابة اسمه في الداخل، ويتم تمثيل الخواص (Attributes) التي تملكها الفئة بصورة متغيرات، ويتم تمثيل السلوك (Operations) التي تقوم بها الفئة بصورة دالة أو طريقة.

أستخدم مخطط الفئات

- ← يستخدم مخطط الفئات في تصميم البرمجيات وتوضيح هيكل النظام وتحديد العلاقات بين الفئات المختلفة وتوضيح متطلبات النظام وتبسيط عملية التطوير والصيانة للنظام.
- ← مخطط يستخدم لهيكل ونمذجة الفئات ومحتوياتها ويعتمد في التصميم على الفئات والكائنات . كما يقوم بتوضيح العلاقات بين الفئات وبعضها .

أنواع العلاقات التي يمكن استخدامها في مخطط الفئات (Class Diagram) لتوضيح العلاقات بين الفئات والكائنات المختلفة في النظام، ومن بين هذه العلاقات:

- 1- التوريث (Inheritance): تستخدم لتوضيح العلاقة بين فئتين حيث ترث الفئة المشتقة (Subclass) خصائص الفئة الأساسية (Superclass).
 - 2- العلاقة التجميعية (Composition): تستخدم لتوضيح العلاقة بين فئتين حيث تحتوي الفئة المركبة (Composite) على كائنات من الفئة المركبة في حين أن الفئة المركبة لا يمكن أن تعيش بدون الفئة التي تحويها.
 - 3- العلاقة التوصيلية (Association): تستخدم لتوضيح العلاقة بين فئتين حيث يحتوي كائن من الفئة الأولى على كائنات من الفئة الثانية.
 - 4- العلاقة التبعية (Dependency): تستخدم لتوضيح العلاقة بين فئتين حيث تعتمد الفئة الثانية على الفئة الأولى.
- وتستخدم هذه العلاقات بشكل مختلف لتوضيح العلاقات بين الفئات والكائنات المختلفة في النظام، وتساعد في تبسيط عملية تصميم النظام وتحديد متطلباته وتحسين جودة النظام.

مكونات الفئة

⇐ الاسم "name" :- ويكون فى الجزء الاول من الفئة ويتم كتابة فيه أسم الفئة .

⇐ الصفات "attributes" :- وتكون فى الجزء الثانى من الفئة ويتم كتابة فيها الصفات اللمميزة للفئة .

⇐ العمليات "operations" :- وتكون فى الجزء الاخير من الفئة ويتم كتابة فيها العمليات التى تقوم بها الفئة

أسم الفئة Class Neme
الخصائص Attributes
العمليات Methods